*mathematics*

MDPI

*Article*

# Reinforcing Network Security: Network Attack Detection Using Random Grove Blend in Weighted MLP Layers

**Adel Binbusayyis**

Department of Software Engineering, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; a.binbusayyis@psau.edu.sa

**Abstract:** In the modern world, the evolution of the internet supports the automation of several tasks, such as communication, education, sports, etc. Conversely, it is prone to several types of attacks that disturb data transfer in the network. Efficient attack detection is needed to avoid the consequences of an attack. Traditionally, manual attack detection is limited by human error, less efficiency, and a time-consuming mechanism. To address the problem, a large number of existing methods focus on several techniques for better efficacy in attack detection. However, improvement is needed in significant factors such as accuracy, handling larger data, over-fitting versus fitting, etc. To tackle this issue, the proposed system utilized a Random Grove Blend in Weighted MLP (Multi-Layer Perceptron) Layers to classify network attacks. The MLP is used for its advantages in solving complex non-linear problems, larger datasets, and high accuracy. Conversely, it is limited by computation and requirements for a great deal of labeled training data. To resolve the issue, a random info grove blend and weight weave layer are incorporated into the MLP mechanism. To attain this, the UNSW–NB15 dataset, which comprises nine types of network attack, is utilized to detect attacks. Moreover, the Scapy tool (2.4.3) is utilized to generate a real-time dataset for classifying types of attack. The efficiency of the presented mechanism is calculated with performance metrics. Furthermore, internal and external comparisons are processed in the respective research to reveal the system's better efficiency. The proposed model utilizing the advantages of Random Grove Blend in Weighted MLP attained an accuracy of 98%. Correspondingly, the presented system is intended to contribute to the research associated with enhancing network security.

**Keywords:** intrusion detection; UNSW–NB15; cyber-attack; machine learning; deep learning; multilayer perceptron; Scapy tool

**MSC:** 68U10; 90C27



**Citation:** Binbusayyis, A. Reinforcing Network Security: Network Attack Detection Using Random Grove Blend in Weighted MLP Layers. *Mathematics* **2024**, *12*, 1720. https://doi.org/10.3390/math12111720

Academic Editor: Miodrag Mihaljevic

Received: 29 April 2024
Revised: 24 May 2024
Accepted: 29 May 2024
Published: 31 May 2024



## 1. Introduction

Globally, with the development of extremely intrusive attacks and the rapid upsurge in cyber traffic, an adaptive and real-time system is paramount for detecting and combating these threats. Cyber-attacks can manifest in different forms, like malware, unauthorized access, or leakage of confidential data [1]. Maintaining a vigilant stance and proactively implementing measures to safeguard individuals, businesses, and industries against such cyber-threats is crucial. Thus, the IDS (Intrusion Detection System) is utilized, in which the IDS activates unceasingly and detects, monitors and responds to emergent threats, empowering organizations to actively protect against cyber-attacks and alleviate the impact of successful breaches. However, prevailing intrusion detection approaches must be improved to handle modern cyber-threats' ever-evolving and intricate nature. Over the past few years, the intrusion detection of networks has become complex due to the constant emergence of new attacks [2]. Accordingly, ML (Machine Learning) has become extensively adopted in IDS, capitalizing on its capability to recognize patterns from complicated data via various algorithms and other statistical approaches. These techniques are particularly

well-suited for mechanical learning, aid in extracting features from widespread datasets, and have demonstrated promising performance [3–7]. Despite these advantages, feature engineering still holds significance in DL models when dealing with high-dimensional structured data. High-dimensionality, inappropriate features and redundant features can potentially lead to overfitting during the learning process.

Correspondingly, a sophisticated hybrid model is implemented in the classical model to improve the security of WSNs (Wireless Sensor Networks) by incorporating intrusion detection [8]. Traditional study employs techniques such as PCA (Principal Component Analysis), and SVD (Singular Value Decomposition), for feature extraction. The use of the excessively conventional Synthetic Minority Technique intends to balance the data, followed by the implementation of the IDS and network traffic categorization. Thus, existing work has evaluated the most effective features acquired from the dataset and validated the models using the DLFFNN (Deep Learning Feedforward Neural Network) approach. Performance analysis is conducted using the original and reduced datasets from NSL–KDD, CICIDS2017, and UNSW–NB15, resulting in better detection rates. A filter-based feature selection technique, MAD (Mean Absolute Difference), is utilized to identify important features from the NSL–KDD datasets and UNSW–NB15 [9]. These important features are then used as input for various ML classifiers, including MLP (Multiple Layers Perceptron), CatBoost, LGBM (Light Gradient Boosting), ETC (Extra-Tree Classifier), RF (Random Forest), and KNN (K-Nearest Neighbors). Finally, the performance of these classifiers is assessed using different metrics, with the LGBM classifier demonstrating results with a better accuracy using the UNSW–NB15 dataset. Similarly, a methodology is developed in the existing model to create a simple yet intelligent security framework for safeguarding the IoT (Internet of Things) from cyber-attacks [10]. This approach combines DRF (Decisive Red Fox) Optimization with DBRF (Descriptive Back Propagated Radial Basis Function) classification. The DRF optimization technique has been utilized for fine-tuning the features needed for precise detection and classification of intrusions, leading to increased training speed and reduced classifier error rates. Furthermore, the DBRF classification model is positioned to differentiate between a regular and an irregular flow of data using optimized features. Finally, the results obtained from this approach are compared with preceding anomaly detection methods using numerous metrics, demonstrating better performance. Several traditional systems are intended to enhance the efficiency of attack detection. Conversely, some are limited in accuracy, have difficulty in handling larger datasets, etc. Accordingly, Figure 1 represents the process involved in the proposed model.

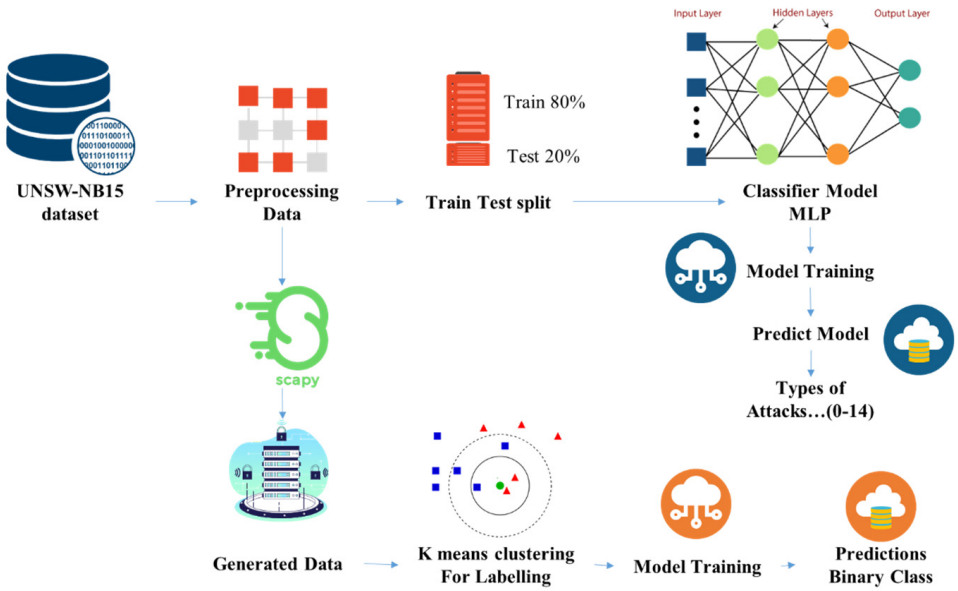



**Figure 1.** Techniques of Respective Mechanism.

To resolve these limitations, the respective system aims to explore and predict cyber-attacks using supervised machine learning algorithms. Initially, UNSW–NB15 and generated real-time datasets are loaded into the system. Then, the pre-processing mechanism with missing values and data normalization is processed to prepare the dataset for the classification. Then, the data is divided in the ratio of 80:20 with 80 parts used for training and 20 parts used for testing the proposed model. Significantly, classification is carried out with Random Grove Blend in the Weighted MLP model. This also involves conducting thorough exploratory data analysis and selecting relevant features to support network IDS. The paper aims to gain insights into attack patterns and characteristics by leveraging machine learning techniques. This knowledge can help organizations detect potential threats early and take proactive measures through robust IDS implementation. Consequently, the main focus of this research is to gather and pre-process the UNSW–NB15 network traffic data. This process involves choosing a suitable model and hyper-parameters, training and evaluating the model, and examining and comparing the outcomes. The major contribution of the proposed system is signified as follows:

- To apply a generated dataset to extract a real-time dataset for classifying types of attacks and a UNSW–NB15 dataset for detecting attacks in the system.
- To utilize the Random Grove Blend in Weighted MLP Layers to enhance the prediction and classification.
- To use performance metrics to calculate the performance of the classification.

### *1.1. Research Questions*

1. How does the combination of Random Grove Blend in Weighted MLP layers improve efficacy and accuracy of network attack detection compared to existing systems?
2. How does the UNSW–NB15 dataset compare to existing public access datasets on the basis of its appropriateness for network attack detection techniques?

### *1.2. Paper Organisation*

This paper begins with an Introduction section briefly describing the present study. In contrast, Section 2 provides a Literature Review that expounds on the major contributions within the relevant research domain. Further, a proposed methodology that describes the research design and procedural framework is deliberated in Section 3. The results in Section 4 include performance analysis, experimental results, comparative analysis, and discussion of results obtained. Section 5 concludes, summarizing the major findings and proposing future work.

## 2. Review of Literature

The section analyzes conventional research on detection of attacks and intrusion detection with deep learning models.

In contemporary research, several advancements have taken place in network attack detection. More complex methods, like ML, DL and AI, are now being used in network attack detection to analyze network traffic patterns, spot anomalies, and find dangers that were previously unidentified. Monitoring user and device activity with behavioral analytic techniques is becoming more common, and integrating these techniques with threat intelligence feeds improves the detection of known hostile entities and activities. Endpoint detection and response (EDR) solutions give granular visibility and device-level reaction capabilities, while cloud-based solutions offer scalability and centralized management of security policies. Processes for responding to incidents are streamlined by automation and orchestration, and security is integrated into DevOps methods to guarantee that security is taken into account at every stage of the software development lifecycle.

In the existing research on attack detection [11,12], several classical models have focused on various techniques to attain better efficiency. For instance, an aided mechanism has been constructed in the traditional model. It comprises significant methods, such as data pre-processing and the function of the CNN (Convolutional Neural Network) model.

Moreover, it is processed with the CSIC2010v2 dataset, which includes 119,585 and 104,000 normal data. It has focused on several types of attacks, such as parameter tampering, XSS, CRLF injection, etc. The prediction result illustrates that the classical system attained an AUC (Area Under ROC Curve) of 0.9696 [13]. Similarly, the conventional method has designed a LDoS (Low-Rate Denial of Service) form of attack detection [14,15]. A multi-feature fusion system and CNN have been developed in the classical model to accomplish this. The outcomes of the classification signify better efficiency [16]. In the same way, a real-time detection system has been constructed based on frequent patterns and flow calculations. The classification has been processed based on the DBN–SVM (Deep Belief Network and Support Vector Machine). In addition, sliding window stream data processing has been developed, and DBN–SVM has been used for the classification. It has functioned with the CICIDS2017 dataset, where the results signify that the traditional model accomplishes better efficacy with better accuracy [17]. Likewise, LSTM (Long Short-Term Memory), MLP, and RF have been utilized in conventional methods to detect cyber-attacks [18,19]. The dataset utilized in the classical system is the CIDDS-001 dataset. The experiment's outcome shows that the LSTM performs with better efficacy than MLP and RF systems and with higher accuracy [20].

Correspondingly, a DDoS (Distributed Denial of Service) attack detection system has been designed in the traditional model [21]. To attain this, the SVC–RF (Support Vector Classifier with Random Forest) classifier has been used in the classification mechanism. This existing model has mainly concentrated on detecting the DDoS attack, which has been processed with the SDN (Situated Dialogue Navigation) dataset. The prediction outcome illustrates the traditional system's better efficacy [22]. An intrusion detection [23,24] system has been designed in the pioneering approach. LSTMgateRNN (Long Short-Term Memory Gate Recurrent Neural Network) has been used to classify attacks. Here, UNSW-SW15, NSL–KDD, and KDD'99 datasets are employed. The experimental outcome signifies that the classical system attained better efficiency [25]. In the same way, the CNN–LSTM-based model has been designed using the conventional method for detecting attacks [26] in wireless sensor networks. It has mainly concentrated on DoS (Denial of Service) attacks. The classification outcome demonstrates that the existing system attained an accuracy of 0.944 [27]. In the same way, wormhole and blackhole attack detection in the wireless sensor networks system has been generated in existing research. This has been processed through the RTT (Round Trip Time) validation mechanism. Here, the LSTM-aided method is utilized for the detection, where the optimal shortest path is defined through the WOA (Whale Optimization Algorithm). In addition, shortest-path communication has functioned in multi-objective functions with metrics such as packet delivery ratio, delay, distance, etc. [28].

Similarly, a large number of conventional models focused on particular types of cyber-attack, such as SQL (Structured Query Language) injection [29–31], Malware attacks [32,33], and phishing attacks [34,35]. Accordingly, the DDoS attack [7] detection model has been constructed in classical research. Here, the detection mechanism uses CNN–O–LSTM. In addition, standard benchmark datasets have been utilized for the classification. The feature selection has been processed with CP–GWO (Closest Position Grey Wolf Optimization) to reduce the similarity between the features [36]. In the same way, attack detection has been processed with the CNN and MLP-based mechanisms. This functions according to the CICDDoS-2019 dataset and the SDN dataset. The experimental results underscore the efficacy of the traditional mechanism [37]. Likewise, the existing method has constructed a DDoS-based attack detection [38,39] system. This has been performed with the Python anaconda platform, along with the OBS_network dataset_2_aug_27.arff. The outcome of the conventional model shows that the pioneering approach attained an accuracy value of 93.54% with the MLPDL (Multilayer Perceptron Deep Learning system) [40]. Congruently, the GOA (Grasshopper Optimization Algorithm) has been utilized in the classical model to detect intrusion in the network. To attain this, the ANN (Artificial Neural Network) has been used to minimize intrusion detection [41,42] error rate. This mechanism functions

by choosing significant parameters, like bias and weights. The outcomes signify the accuracy of the classical system to be 95.41% [43]. Accordingly, a DDoS attack detection system has been constructed in the classical model. To accomplish better efficacy in attack detection [44–46], it has incorporated the CNN–LSTM mechanism, processed with the KDD dataset. The classification outcome represents better detection than the existing system [47]. Correspondingly, a classification network has been designed in the traditional mechanism with the NSL–KDD and KDD–CUP 99 datasets.

Moreover, common DL-based systems, such as GRU–RNN, LSTM–RNN, DBN, and DNN models, have been compared. The result of the conventional system shows better efficacy in network intrusion detection [48]. In the same way, an RNN (Recurrent Neural Network)–CNN based system has been constructed in the traditional approach. The experimental results demonstrate that the conventional model attains an accuracy of 0.975 in its intrusion detection mechanism [49]. In the existing research, a DL-based intrusion detection system has been designed with the KDD CUP 1999 dataset. A CNN-based system has been utilized to detect DoS attacks in order to attain this. Here, the system has generated two types of intrusion images: grayscale and RGB (Red, Green, and Blue). In addition, experimental results represent better efficiency, with 91.5% in CSE–CIS–IDS 2017 and 93% in multiclass classification [50]. Congruently, SVM, MLP, and RF-based mechanisms have been designed in the classical model to detect DDoS attacks. The prediction outcome represents better attack detection efficiency, whereas the MLP technique attained a better accuracy of 97.96% [51].Similarly, the LSTM–RNN (Long Short-Term Memory and Recurrent Neural Network) system has been generated to classify network attacks. Here, attack detection has been processed with the NSL–KDD dataset, which comprises four types of attacks with 2931 instances. The results of this traditional system demonstrate that binary classification achieved a testing accuracy of 95.94%, and multi-class classification achieved a testing accuracy of 82.06% [52].

*Research Gaps*

- The classical study develops an IDS by analyzing the UNSW–NB15 dataset and using four classification models but is deficient in accuracy metrics [53].
- The conventional system utilized DL-based technique for multi-class classification in intrusion detection but is deficient in handling diverse datasets [54].

With the challenges found in the existing research [53], the proposed system suggest a thorough approach to improve intrusion detection systems (IDS) and classification models.

The traditional research, mentioned in citation, focuses on creating an IDS by studying the UNSW–NB15 dataset and utilizing four classification models. Even though this method is a great advancement, it is important to point out that the research does not include specific accuracy measurements. Lack of comprehensive accuracy assessment makes it difficult to grasp the IDS's performance and reliability, which could impede its efficacy in real-world scenarios.

Additionally, the traditional system referenced [54] utilizes a deep learning (DL) method for multi-class classification in intrusion detection. Although DL techniques show potential for managing intricate data, this system has been reported to face challenges with varied datasets. A system's capacity to accurately detect and categorize intrusions in changing network environments can be compromised if it cannot properly manage data variations, thus reducing its usefulness in real-world scenarios.

Based on these findings, our proposed approach aims to address these research gaps by incorporating reliable accuracy measures and improving the versatility of classification models for various datasets. Our goal is to create an IDS framework that provides better performance, reliability, and scalability in actual intrusion detection situations by tackling these main constraints.

## 3. Proposed Methodology

The learning model undergoes training and testing on identical data categories in a conventional ML assessment approach. During the training phase, the model acquires the ability to recognize patterns exhibited by each data category. Subsequently, in the testing phase, the model utilizes learned patterns to identify data samples from the same data categories encountered during training. In the proposed MLP-based IDS system context, the model is trained and tested using predefined and manually generated attack categories. Therefore, the evaluation of the model focuses on its efficiency in detecting malicious data samples originating from both multi-class and binary classification. The overall process of the presented methodology is illustrated in Figure 2.

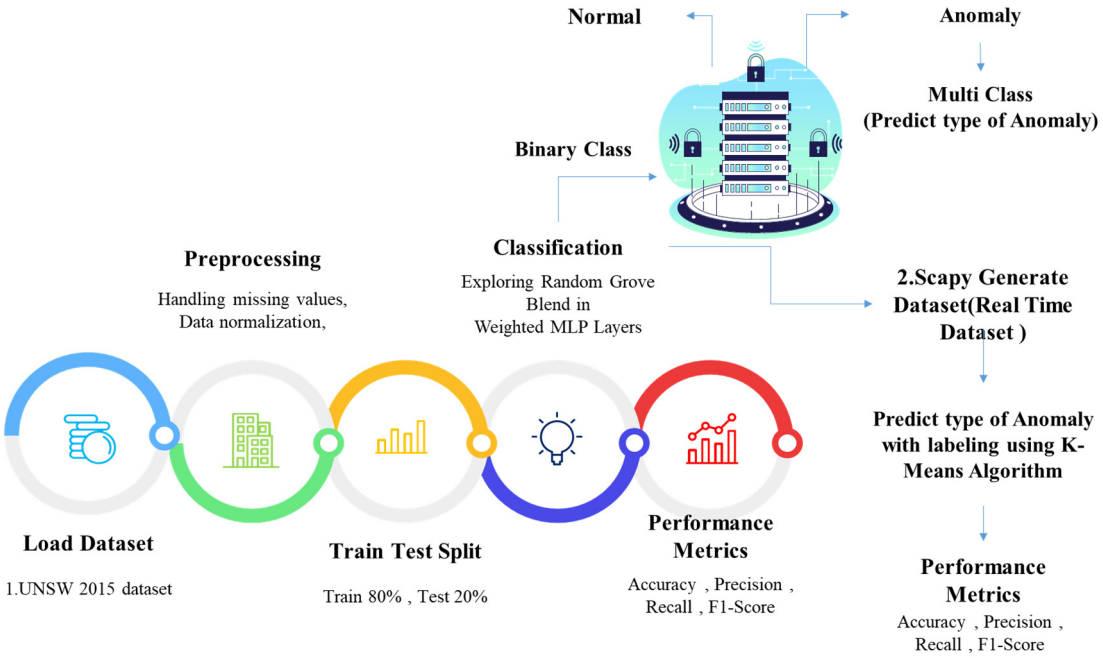



**Figure 2.** Design Flow of the Proposed Model.

Figure 2 illustrates that the mechanism of the presented system comprises data selection, pre-processing, data splitting, and classification. Accordingly, classification in the proposed mechanism comprises two types of classes, binary and multi-class. Normal and anomaly classes are predicted in binary classification, whereas the multi-class classification predicts anomaly types. The precise explanation of the respective systems is depicted in the following sub-sections.

### *3.1. Data Selection*

The proposed system utilized two datasets for the classification: UNSW–NB15 and generated real-time datasets.

#### 3.1.1. UNSW–NB15 Dataset

The UNSW–NB15 dataset is generated in the IXIA perfect storm tool in the cyber range lab of the ACCS (Australian Centre for Cyber Security). It combines synthetic contemporary attack behaviors from network traffic and realistic modern normal activities. Consequently, 2,540,440 pieces of data are stored in the four CSV (Comma Separated Value) files. The tcpdump tool captured 100 GB of raw traffic, fic, Bro-IDS, and Argus to produce 49 features with class labels. Moreover, the number of training and testing sets recorded comprises 175,341 and 82,332 from diverse types, i.e., normal and attacks. Table 1 represents the types of attacks and their descriptions.

**Table 1.** Types of Attacks in UNSW–NB15 Dataset.

| S.No | Attack Types | Description |
|---|---|---|
| 1. | Fuzzers | Attackers send random data to identify liabilities in the system. |
| 2. | Analysis | Attackers collect data related to the system to plot attacks. |
| 3. | Backdoors | Attackers generate concealed access points to operate the system remotely. |
| 4. | DoS | Attackers process traffic to make the system inaccessible to users. |
| 5. | Exploits | Attackers use system vulnerabilities for illegal access. |
| 6. | Generic | Attackers function against each block cyber with a block function with a hash function. |
| 7. | Reconnaissance | Attackers collect data on system vulnerabilities. |
| 8. | Shellcode | Attackers insert mischievous code to perform commands in the system. |
| 9. | Worms | It is a self-replicating attack that spreads across networks. |

Correspondingly, the dataset comprises the following factors:

- Features
- Basic features
- Content features,
- Time features
- Additional genera
- Ted features

Dataset Availability: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 20 February 2024).

#### 3.1.2. Scapy Generated Dataset

The proposed system utilizes the Scapy tool to generate real-time datasets to classify attacks. The definitive interactive packet manipulation tool examines, captures, sends, and produces e-network packets. In addition, it supports the simulation of the diverse types of attacks in the network. The respective research generated a binary class for the classification of network attacks. In addition, it is a powerful tool for modifying and manipulating packets. It is a library built on Python software (3.7) and provides effective capabilities for creating, transferring, capturing, and analyzing network packet traffic at different levels of the network stack. Scapy can interpret packets from various protocols and perform common tasks, such as network discovery, scanning, attacks, unit testing, probing, and tracing routes. One of the notable features of Scapy is its built-in packet dissection engine, which allows for the decoding of a wide range of network protocols, including Ethernet, IP (Internet Protocol), TCP (Transmission Control Protocol), and UDP (User Datagram Protocol). This feature provides great flexibility and versatility in analyzing network traffic. It is worth mentioning that Scapy can also be used for malicious activities, as it allows for packet manipulations. Therefore, this work utilizes Scapy as a traffic generator for generating packets and spoofing the source IP address of these packets using the k-means clustering technique.

### *3.2. Pre-Processing*

The input data from the dataset is a function with a specific pre-processing mechanism to formulate the dataset for predicting and classifying attacks. Consequently, the presented method utilized two pre-processing techniques: handling missing values and data normalization.

3.2.1. Handling Missing Values

It recognizes and addresses the dataset's null or incomplete points, which are processed by removing columns or rows with missing values. In addition, it processes missing values through statistical measures for identifying and filling in these missing values.

3.2.2. Data Normalization

The data normalization includes scaling and transforming the data to the standard range between 0 and 1 or $-1$ and $-1$. It alters the numerical values in the dataset to utilize them in the common scale. The dataset exhibits a significant contrast between its maximum and minimum values. To simplify the algorithm's processing, it is necessary to normalize the data. Data normalization allows for optimal utilization of classification algorithms. For instance, if the back-propagation technique is employed in a multilayer perceptron (MLP), the normalization of input values can accelerate the training phase, resulting in a more efficient network. The primary normalization method is data scaling, specifically the minimax algorithm. This algorithm can convert the current data range to a standardized interval, typically $[-1, 1]$ or $[0, 1]$. The mathematical formula of normalization is presented in Equation (1).

$$n = \frac{(y - y_{min})(max - min)}{(y_{max} - y_{min})} + min \quad (1)$$

where $n$ is the converted input value, ($y_{max}$ and $y_{min}$) represent the initial range of the input variable's values, and ($max$) signifies the specified range of the input variable.

Pre-processing is a highly important function, making sure that missing values re dealt with and normalizing the data. Using a combination of removal and imputation approaches, missing values were handled while paying close attention to preserving dataset integrity. By using scaling techniques, like MinMax scaling, to ensure uniform feature scales and lessen the effect of outliers, data normalization was accomplished. Furthermore, the dataset was enhanced with significant features by the application of feature selection techniques and the proper encoding of categorical variables. In order to discover and lessen the impact of outliers on model training, outlier detection techniques were put into place. The capacity of these pre-processing processes to guarantee robustness in the face of changing data features and enhancing model performance made them warranted. SHAP values were determined to comprehend the influence of many aspects.

### *3.3. Data Splitting*

The pre-processed data are further divided into training and testing based on 80% and 20%. This is used to train the classification system and evaluate its performance. In addition, it is utilized to avoid overfitting of data. Using 80% of the data for training allows the model to learn complex patterns, while reserving 20% for testing ensures that the model's performance can be accurately assessed on unseen data.

### *3.4. Classification: Random Grove Blend in Weighted MLP Layers*

The proposed system used the RGBW–MLP System to predict and classify network attacks. MLP is considered a widely used NN (Neural Network) model, which is applied for classification and regression problems. In addition, MLP comprises multiple layers of neurons connected by weighted edges. This approach tends to learn the weights of the edges by reducing an objective function using back propagation. Accordingly, it handles complete handler problems, larger datasets, and high accuracy. Conversely, it is limited through computation and requirements for a great deal of labeled training data. A random Info Grove Blend and weight weave layer are incorporated into the MLP mechanism to address the issue. Correspondingly, by adding randomness to the data processing pipeline, the Random Info Grove Blend layer broadens the representation of the data. Neural network approaches are deliberately combined with this randomness to improve the model's stability and generalization across a variety of input data circumstances. The

model learns more about the input space and performs better in classification when random data is added to the regular neural network processing.

Figure 3 shows the Random Grove Blend in Weighted MLP Layers.

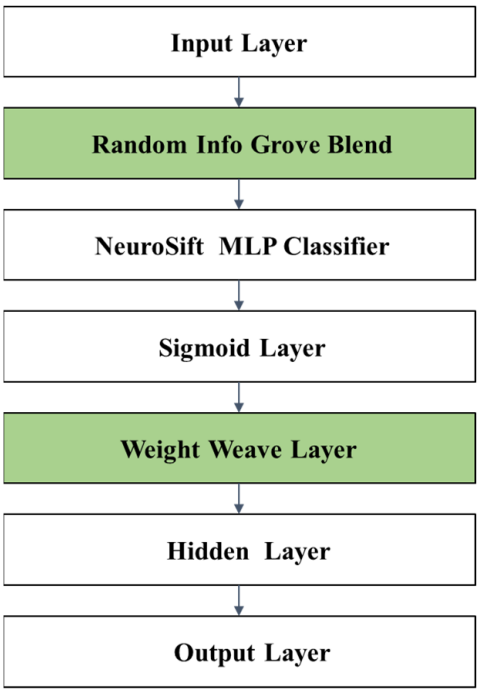



**Figure 3.** Random Grove Blend in Weighted MLP Layers.

The modified MLP layers incorporating Random Info Grove Blend and weight weave layer are represented in Figure 3. The explanation of the layers in the proposed model is depicted below:

- *Input Layer:* The input layer receives and passes the input data to the Random Info Grove Blend.
- *Random Info Grove Blend:* This processes the input data by incorporating random data and neural network techniques. It is utilized to improve the classification performance in processing the diversity in the input data.
- *NeuroSift MLP Classifier:* The MLP classifier extracts features from the input data for a functioning prediction mechanism in the layer. The process includes sifting through the data to recognize significant patterns.
- *Sigmoid Classifier:* The sigmoid classifier is the activation function that presents non-linearity in the neural network. It handles complex data relationships, enhancing the classification's interpretability.
- *Weight Weave Classifier:* This layer combines the weight learned in the neural network for the prediction mechanism. It is processed by learning neural networks for the prediction of attacks. A novel feature of our model, the Weight Weave Layer incorporates a complex regularization process to improve the model's learning capabilities. This layer intertwines weights in a way that enforces sparsity, reduces redundancy, and saves key information in an effort to avoid overfitting and promote generalization.
- *Hidden Layer:* This is the intermediate layer that functions with the extracted data. In addition, it processes compound computations to alter the input data to the format needed for the classification.
- *Output Layer:* The final layer generates the outcome of the classification. It produces final predictions regarding processed data and learned weights from the preceding layers.

Correspondingly, the architecture of the projected system is depicted in Figure 4.

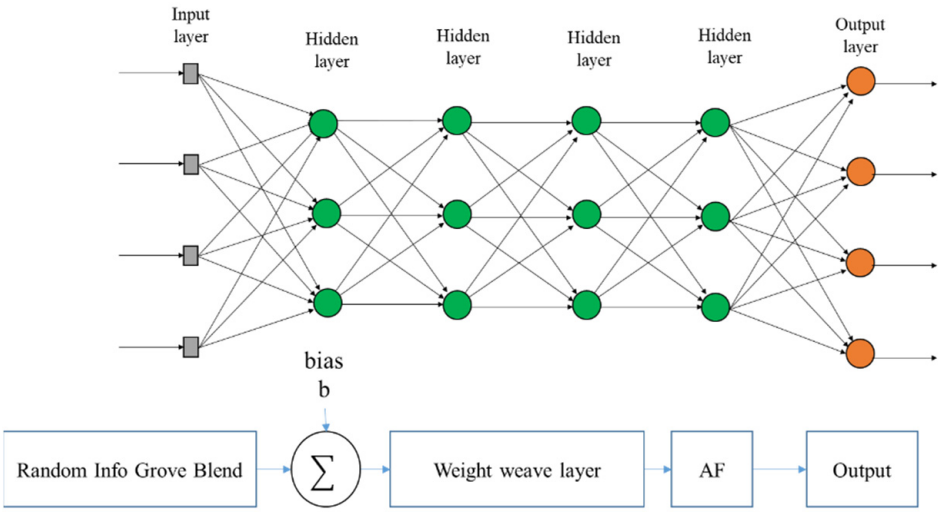



**Figure 4.** Architecture of the Proposed Method.

The attention in the respective research calculates the alignment score from two sources. The input FC adjacency sequence is depicted in Equation (2).

$$M = [M_1, M_2, \ldots M_d]\ query\ \in R^d \tag{2}$$

Correspondingly, the initial layer receives the input data, where the outcome is the weighted nonlinear function of each input. The initial stage outcome is represented in Equation (3).

$$M^{(1)} = b_0 + W^{(0)} f^{(0)}(x) \tag{3}$$

Here, $x$ signifies the genotypes of the individual. In addition, $b_0$ depicts bias, assessed with the rest of the weights, $W^{(0)}$. Moreover, $f$ represents the nonlinear function. In the subsequent layers, a similar function is utilized for the classification. Conversely, neuron inputs in the given layer are processed from the preceding layer $M^{(k-1)}$, illustrated in Equation (4).

$$M^k = b_{k-1} + W^{k-1}\ f^{k-1}\left(M^{(k-1)}\right) \tag{4}$$

Accordingly, the softmax function is used, which alters the alignment scores. It is shown as $\left[f(M,\ Q)\right]_{i=1}^{d}$, processed with the probability distribution function $p(z|M,\ Q)$. Here, $z$ illustrates an indicator where the element is significant for $Q$. Congruently, $p(M$ and $Q)$ signifies that $M$ supports the extraction of the significant features to $Q$. The equation shows the attention mechanism (5).

$$softmax = [f(A, Q)]_{i=1}^{d} \text{ and } p(M, Q) = softmax(\alpha) \tag{5}$$

The outcome $y_i$ represents the weighted element based on the significance of the proposed classification, which is depicted in Equation (6).

$$y_i = p(M, Q)A \tag{6}$$

Likewise, the compatibility function in the attention mechanism is signified as $f(.)$, which is parameterized by the proposed MLP function.

$$f(A, Q) = wgt^T \sigma(wgt^{(1)} M_i + wgt^{(2)} q) \tag{7}$$

Here, learnable parameters are represented as $wgt^{(1)} \in R^{DxD}, wgt^{(2)} \in R^{DXD}$, and $wgtR^D$, where the dimension is $D$, which is the dimension of $A$. Here, the attention system

uses the inner product as the compatibility function for $f(A,\ Q)$. This is illustrated in Equation (8).

$$f(A, Q) = \langle wgt^{(1)} A,\ wgt^{(2)} Q \rangle \tag{8}$$

Consequently, $Q$ is detached from the compatibility function, which is represented in Equations (9)–(11).

$$F(A) = wgt^{T} \sigma(wgt^{(1)} A) \tag{9}$$

$$\alpha = [f(A, Q)]_{i=1}^{dim} \tag{10}$$

$$p(x, Q) = softmax(\alpha) \tag{11}$$

Finally, the outcome of the weighted element is $o_i$, which functions based on the importance of the classification. The outcome is depicted in Equation (12).

$$o_i = p(M)A\ *\ y_i^{'} \tag{12}$$

Correspondingly, the proposed classification mechanism is evaluated using the performance metrics. The end result of the classification is determined by the data that has been processed and the weights that have been learned from the previous layers. By utilizing strong feature extraction and regularization methods, the system generates forecasts about network attacks.

The FC adjacency sequence is subject to different alterations and actions across the layers, as illustrated in Equations (3)–(12). The Random Info Grove Blend layer is essential for expanding the variety of data representation and improving the model's resilience and generalization ability. The RGBW–MLP system, which is combined with the Weight Weave Layer, effectively overcomes the drawbacks of traditional MLPs, resulting in enhanced prediction and classification abilities, especially in network attack scenarios. By utilizing these innovative layers, the structure showcases excellent performance metrics.

Uniqueness of the Random Grove Blend Layer

*Key innovations:*

Improved model stability: The addition of a processing pipeline improves model resilience by mimicking different data situations, enhancing overall performance.

1. Utilizing sophisticated neural network techniques in the fusion layer enables effective management of varied and intricate input data.
2. Enhanced Classification Performance: The Random Grove Blend layer enhances classification accuracy and performance by diversifying data representation, particularly in predicting network attacks.
3. These improvements make the Random Grove Blend and Weight Weave Layer key advancements in the RGBW–MLP system, enhancing its efficiency and dependability in predicting and classifying network attacks.

## 4. Results and Discussion

The section explores the viability of the projected MLP model by presenting the valuable results obtained in detecting malicious attacks in the network. In addition, it deliberates the performance analysis of the proposed MLP model with other state-of-the-art methods to verify the efficiency of the respective system.

### *4.1. EDA (Exploratory Data Analysis)*

The EDA is used to overview and examine the data extracted from the dataset. Here, EDA is functioned with the explainable AI, with SHAP (SHapley Additive exPlanations) values. This is a method used within the presented model to explain the output by attributing the impact of each feature on the prediction. It breaks down predictions to illustrate how each feature influences the model's outcome, offering insights into feature importance

and model behavior. Accordingly, Figure 5 signifies the mean SHAP value in terms of each feature.

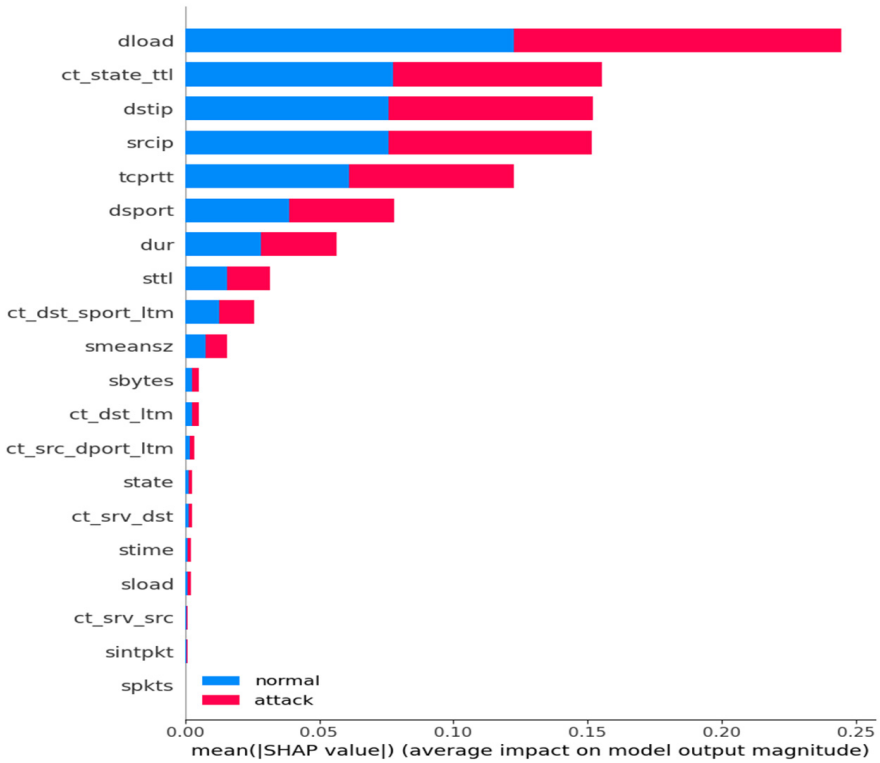



**Figure 5.** Mean SHAP Value.

Figure 5 represents the average impact of normal and attack data in the utilized dataset in terms of SHAP mean value and in terms of normal and attack. In addition, it is used to analyze the influence of every feature in the prediction system. It allocates significant values to every features in the system. Positive SHAP values will positively influence the prediction and negative SHAP values will negatively influence the prediction. Hence, the presented model depends on the features with positive and negative SHAP values Here, it is identified that the dload is comprised of the maximum data and spkts the minimum amount of data. Correspondingly, Figure 6 represents the SHAP value. The equation for Mean SHAP value is represented in Equation (13).

$$\text{Mean SHAP value}: \frac{1}{N}\sum_{i=1}^{N} \text{SHAP value}_j^i \tag{13}$$

Here, the total number of of instances in the dataset is signified as $N$ and the attack feature which the proposed system calculates is shown as $j$. significantly, SHAP value for the feature $j$ for the ith instance in the utilized dataset is depicted as SHAP value$_j^i$. This formula calculates the average SHAP values of the attack feature among all cases in the data to determine its overall significance or influence on the proposed model's prediction.

Figure 6 illustrates the SHAP value on the basis of features with positive and negative influence on the outcome of the presented model. It is computed by relating the prediction to and without each feature in the dataset for the purpose of representing the feature influence on the proposed model. The equation of SHAP is represented in Equation (14).

$$\varnothing_i = \sum_{s \subseteq N} \frac{|S|!(|N| - |S| - 1)!}{|N|!}[f(S \bigcup \{i\}) - f(S)] \tag{14}$$

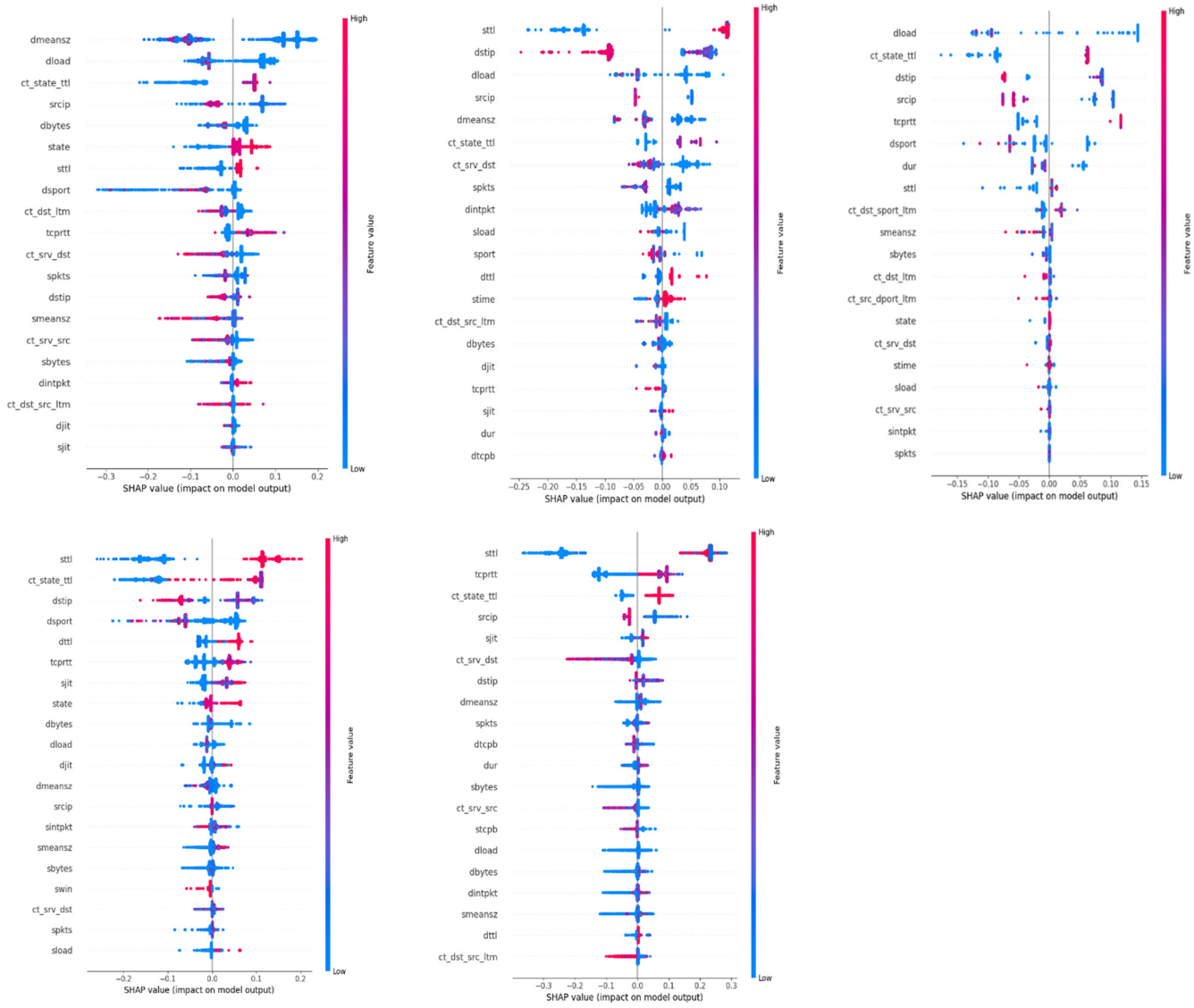



**Figure 6.** SHAP value.

Here, the whole features set is shown as $N$, where the number of whole features is represented as $|N|$, and the SHAP value of feature i is signified as $\varnothing_i$. In addition, the subset of features that do not comprise feature I is presented as $S$, where the number of features in subset $S$ is displayed as $|S|$ and the prediction function of the presented system is signified as $f$.

### *4.2. Performance Metrics*

The efficacy of the respective research is calculated with performance metrics such as F1 score, Accuracy, Recall, and Precision.

1. *F1-score:* The F1-score is figured by computing the mean of recall outcome and precision outcome, which is described in Equation (15)

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (15)$$

2. *Accuracy:* The accuracy is the vital metric that is figured by captivating the ratio of correct identification to complete identification. The formula for accuracy is shown in Equation (16)

$$Accuracy = \frac{Tp + Tn}{Tp + Fp + Tn + Fn} \tag{16}$$

where $Tn$ is True Negative, $Tp$ is True Positive; $Fn$ is False Negative, and $Fp$ is False Positive

3. *Recall:* The recall is the ratio of correctly identified values to the whole of the identified values. It is also termed as specificity or sensitivity, which is indicated in Equation (17)

$$Recall = \frac{Tp}{Tp + Fn} \tag{17}$$

where $Fn$ and $Tp$ are False Negative and True Positive.

4. *Precision:* The precision is the computation of the recognized positive figure. It is calculated by the fraction of true positives to the average of true positives and false positives. It is demonstrated in Equation (18)

$$Precision = \frac{Tp}{Tp + Fp} \tag{18}$$

where $Fp$ is False Positive and $Tp$ is True Positive.

- $Tn$: Denotes the number of cases successfully detected as non-attack and classified as such (True Negatives).
- $Tp$: Denotes the number of cases correctly detected as attacks that are classed as such (True Positives).
- $Fn$: Indicates the number of cases that were wrongly labeled as non-attacks but were nonetheless classed as attacks (False Negatives).
- $Fp$: Indicates the number of cases that were wrongly classified as attacks but were really classified as non-attacks (False Positives).

### *4.3. Experimental Results*

Following this, Table 2 represents the multi-class classification output of the projected MLP model with the UNSW–NB15 dataset. In addition, a graphical representation of the results obtained is signified in Figure 7.

**Table 2.** Multi-class Classification Results.

| Multi-Class | Precision | Recall | F1-Score |
|---|---|---|---|
| 0 | 0.56 | 0.57 | 0.57 |
| 1 | 0.48 | 0.5 | 0.49 |
| 2 | 0.61 | 0.78 | 0.68 |
| 3 | 0.26 | 0.55 | 0.36 |
| 4 | 1 | 0.99 | 0.99 |
| 5 | 0.55 | 0.38 | 0.55 |
| 6 | 0.26 | 0.25 | 0.36 |
| 7 | 0.26 | 0.26 | 0.69 |
| 8 | 0.69 | 0.21 | 0.38 |
| 9 | 0.89 | 0.56 | 0.69 |
| 10 | 0.98 | 1 | 0.99 |
| 11 | 0.7 | 0.78 | 0.74 |
| 12 | 0.29 | 0.44 | 0.35 |
| 13 | 0.78 | 0.69 | 0.38 |
| Weighted avg | 0.98 | 0.98 | 0.98 |
| Accuracy | 0.98 | | |

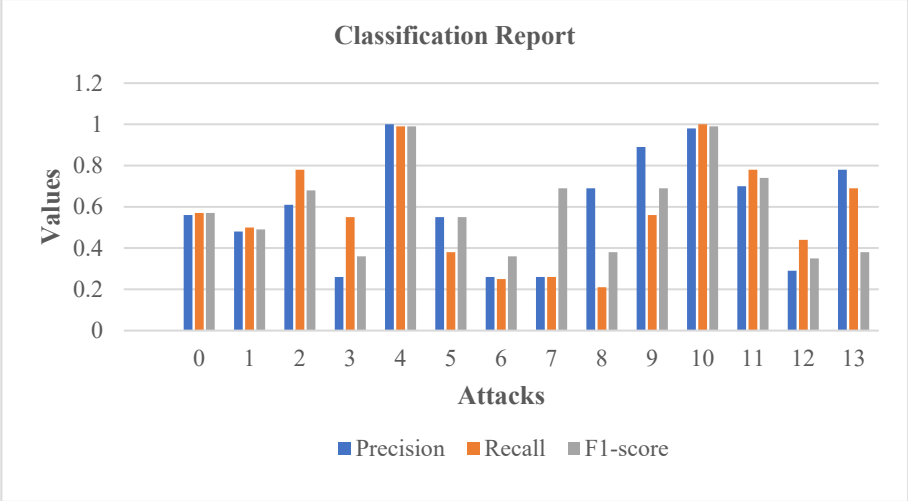



**Figure 7.** Graphical Representation of Multi-class Classification Outputs.

From analysis, it is identified that the proposed MLP model produced an accuracy of 0.98 and improved precision, recall, and F1-score values. This denotes the better classification of multi-class anomalies using the proposed MLP combined with Random Info Grove Blend and weight weave layers. Further, the classification results of binary classes with the Scapy tool are denoted in Table 3.

**Table 3.** Binary Classification Outputs.

| Binary Class | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

The binary classification is performed using the proposed MLP model with a dataset generated by the Scapy tool. It is identified that the accuracy, precision, recall, and F1-score is 1, which denotes that the proposed MLP model achieves an improved detection rate in binary classification. The confusion matrix for multi-class classification obtained with UNSW–NB15 is demonstrated in Figure 8.

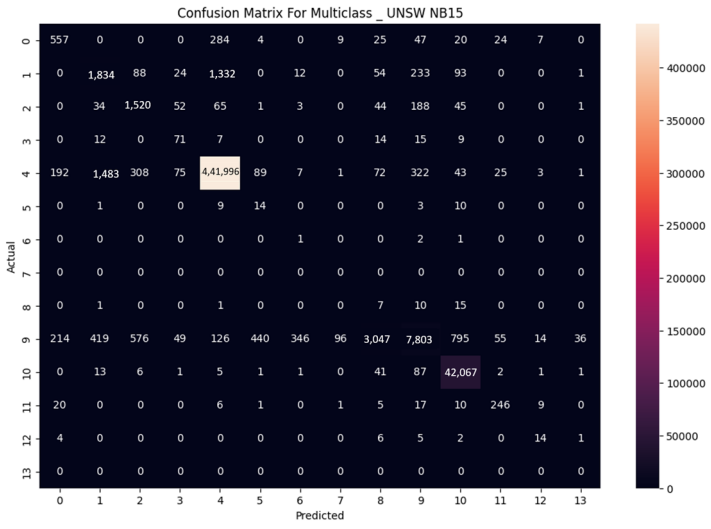
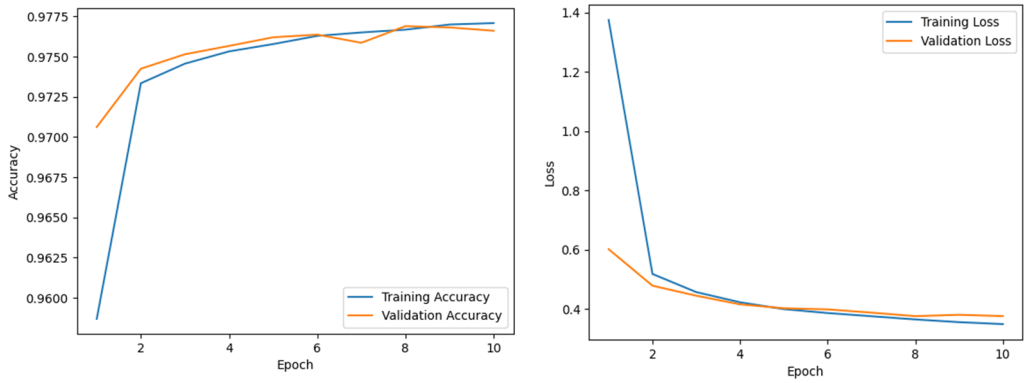



**Figure 8.** Confusion Matrix for Multiclass UNSW–NB15.

A confusion matrix signifies the classification, associating its forecasted labels to the true labels. It shows the number of TPs (True Positives), TNs (True Negatives), FPs (False Positives), and FNs (False Negatives) of the model's predictions. As shown in Figure 8, the confusion matrix for multi-class classification deliberates that only minimum error has been obtained while classifying 14 different classes of attacks. Further, the model accuracy and loss graph for multi-class classification are shown in Figure 9.




(**a**) Accuracy (**b**) Loss

**Figure 9.** Accuracy and Loss Curve.

The improvement in accuracy with reduced loss and error rate signifies the enhanced performance of the proposed MLP model. In Figure 9, the training curve (blue) indicates if the model is learning, and the validation curve (orange) denotes if the model can generalize well for new data. In this segment, an evaluation depends on the study's findings on the proposed MLP model for the recognition of attacks present in the network. The loss and accuracy procured with training data are performed for effective comparison and to analyze the effectiveness of the proposed MLP model. Further, the confusion matrix for binary classification with Scapy is signified in Figure 10.

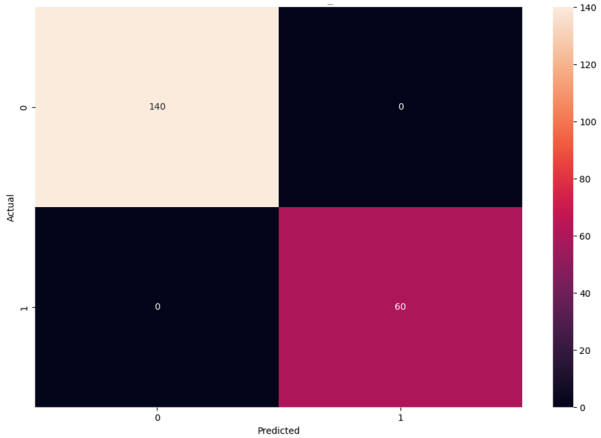
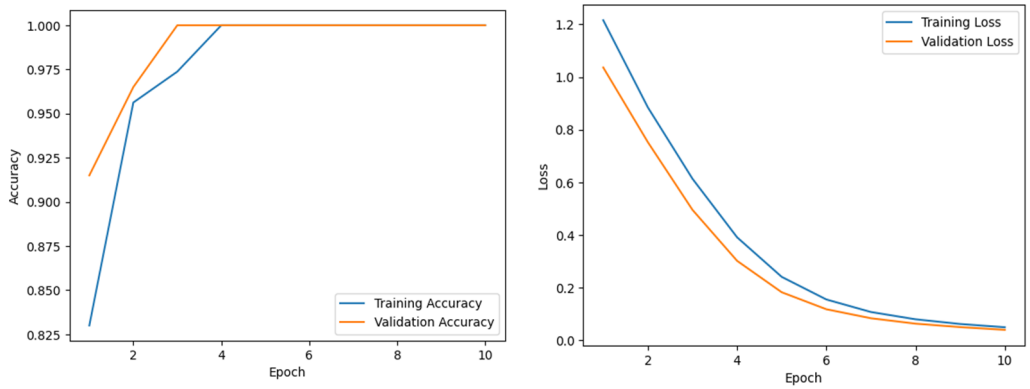



**Figure 10.** Confusion Matrix for Binary Classification.

Figure 10 shows that the confusion matrix for binary classification using the Scapy dataset produces an improved detection rate. The binary classification for attack and normal classes is carried out, where normal classes are correctly classified with 140 samples. Meanwhile, attack classes are classified correctly with 60 samples. Further, the model accuracy and loss graph for binary classification are shown in Figure 11.




(**a**) Accuracy (**b**) Loss

**Figure 11.** Accuracy and Loss Curve.

Generally, the training and validation learning curve produced with noise typically produces a high loss, denoting that the model cannot learn the training dataset. In the case of a loss curve with training data, the training loss and validation loss are identified as low. This indicates that the model's performance is improved with less loss rate. By evaluating the accuracy, the performance of the model is easily predicted. This denotes the count of predictions where the predicted value equals the original value.

Nevertheless, the accuracy is a parametric component that shows an optimistic response if the input data presents any bias. The present study measures the accuracy of validation data with each training data count. From analysis, it is identified that, in the graphs in Figure 11, an accuracy rate of 80% training data is detected and improves as the number of samples increases.

### *4.4. Comparative Analysis*

The effective comparison of the proposed classification model with existing approaches is evaluated in this section. The results procured by existing and proposed MLP methods are shown in Table 4, and the corresponding graphical representation is depicted in Figure 12.

**Table 4.** Comparison of Proposed Results with Existing Methods [55].

| Models | Accuracy |
|---|---|
| Existing method | 97.37 |
| Proposed method | 98 |

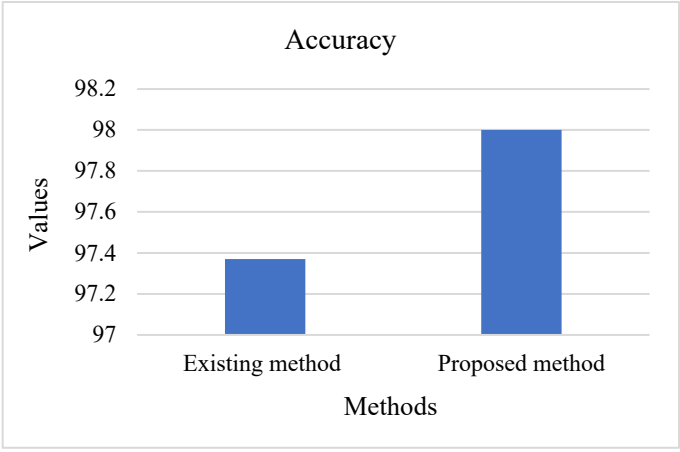



**Figure 12.** Graphical Representation of Comparative Results.

The accuracy of the suggested approach was 98%, compared to 97.37% for the current method. Using the existing method, Table 3 and Figure 12 identify that the accuracy of the proposed system attained a score 0.63 higher than the existing model. This suggests that, when compared to the current system, the suggested method is more accurate in correctly classifying occurrences. The suggested method's increased accuracy raises the possibility that it could be more successful in precisely recognizing and categorizing incidents, which is important when it comes to attack detection. This outcome reveals the better efficacy of the presented approach. Figure 13 and Table 5 show the comparative analysis of the respective model with the traditional method.

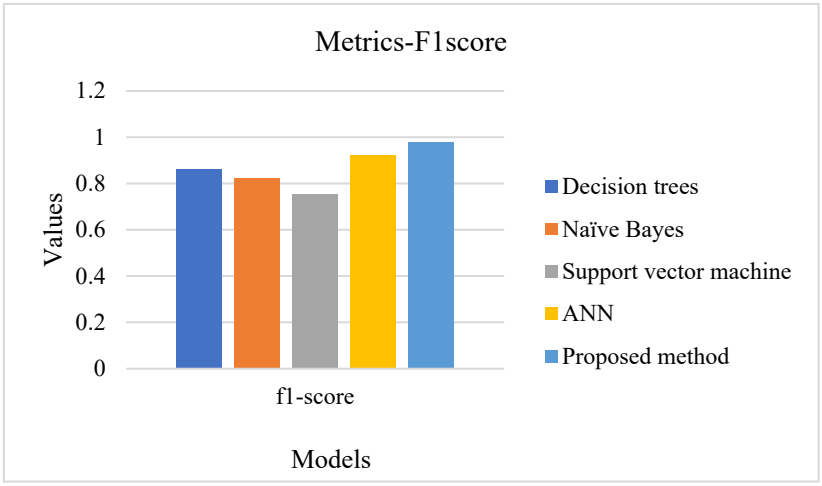



**Figure 13.** Graphical Representation of Comparative Results.

**Table 5.** Assessment of Proposed Outputs with Conventional Models [56].

| Methods | F1-Score |
|---|---|
| Decision trees | 0.86 |
| Naïve Bayes | 0.824 |
| Support Vector Machine | 0.755 |
| ANN | 0.92018 |
| Proposed Method | 0.98 |

The analysis shows that the existing model's ANN is higher among the conventional systems with a 0.92 f1-score. However, the respective method attains 0.98, which is a significantly 0.6 value greater than the classical model, showing the better efficiency of the respective research. This higher F1-score of the presented method suggests that it is more effective than the conventional models in correctly identifying and categorizing occurrences while limiting false positives and false negatives. It also shows excellence in precision and recall. Accordingly, Figure 14 and Table 6 compare the outcome with traditional results.

The proposed model shows a notable increase in accuracy in comparison to the traditional techniques. In detail, the respective method obtained an accuracy level of 0.98, while Zhang et al. reached 0.8599, DBN achieved 0.82, CDBN reached 0.8229, and the existing method reached 0.8649. The proposed method surpasses Zhang et al. [57] by around 11.01%, DBN by about 16.83%, CDBN by roughly 16.71%, and the current method by approximately 13.51% when comparing these values. This significant enhancement in precision demonstrates the superiority of the suggested model in accurately categorizing instances when compared to other methods. The suggested model's high accuracy implies its reliability in detecting attacks effectively, which is essential for maintaining network security. The comparison results depict that the proposed model reaches a maximum of 0.16 and a minimum of 0.1151 higher than the classical model. This exposes the better

performance of the presented approach. Consequently, Figure 15 and Table 7 compare the respective systems with existing methods.

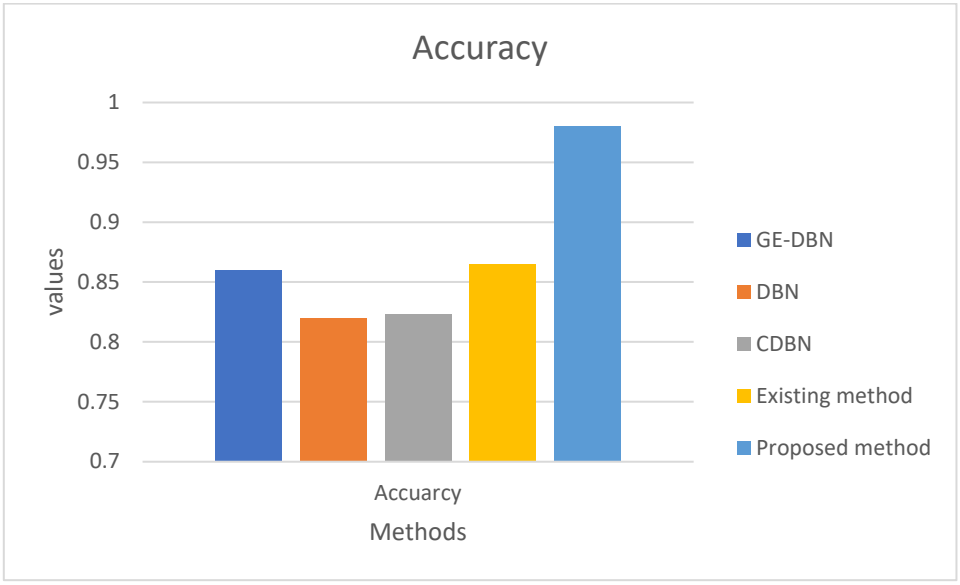



**Figure 14.** Graphical Representation of Comparative Results [57].

**Table 6.** Comparison of Proposed Results with Existing Methods [57].

| Methods | Accuracy |
|---|---|
| GA-DBN. | 0.8599 |
| DBN | 0.82 |
| CDBN | 0.8229 |
| Existing method | 0.8649 |
| Proposed Method | 0.98 |

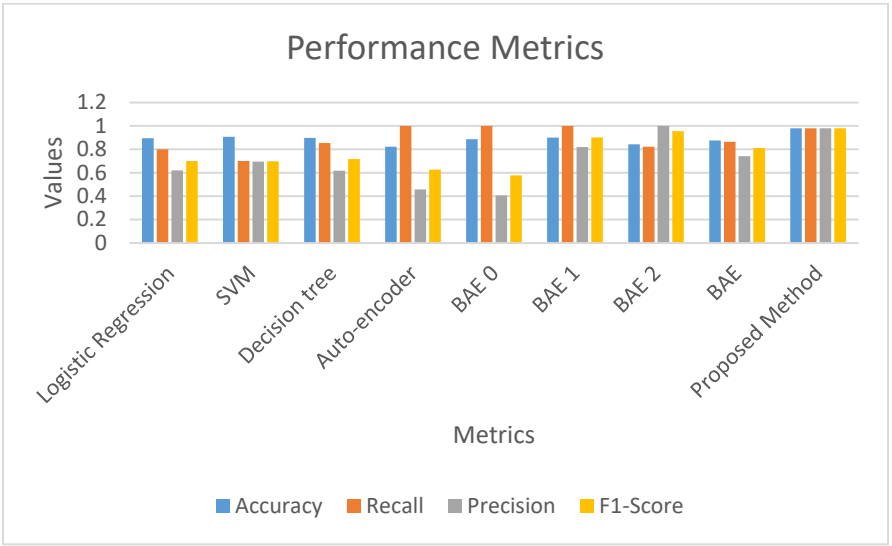



**Figure 15.** Graphical Representation of Comparative Outcomes.

**Table 7.** Comparison of Proposed Results with Traditional Outcomes [58].

| Methods | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 0.8951 | 0.8 | 0.62 | 0.7004 |
| SVM | 0.9075 | 0.7011 | 0.6949 | 0.698 |
| Decision tree | 0.897 | 0.8541 | 0.6173 | 0.7166 |
| Auto-encoder | 0.822 | 1 | 0.4566 | 0.6269 |
| BAE 0 | 0.886 | 1 | 0.4063 | 0.5778 |
| BAE 1 | 0.90002 | 1 | 0.8198 | 0.901 |
| BAE 2 | 0.8425 | 0.8229 | 1 | 0.955 |
| BAE | 0.8761 | 0.8649 | 0.742 | 0.8113 |
| Proposed Method | 0.98 | 0.98 | 0.98 | 0.98 |

In the comparative analysis, it is recognized that the proposed system accomplished a maximum of 0.16 accuracy, 0.18 recall, 0.37 precision and 0.41 f1-score greater than the conventional models. The outcome of the comparative analysis depicts that the proposed system attains better efficiency than the existing system, with 0.98 in all four performance metrics. This is significantly higher than the conventional system. Therefore, a comparative analysis of the projected system with the classical approach exposes the greater efficiency of the respective research. Table 8 and Figure 16 demonstrate the comparative analysis of the proposed model with traditional systems.

**Table 8.** Comparative Analysis of Presented System with Existing Method.

| Methods | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| LSTM | 0.9611 | 0.97 | 0.95 | 0.96 |
| GRU | 0.9698 | 0.96 | 0.97 | 0.97 |
| Proposed method | 0.98 | 0.98 | 0.98 | 0.98 |

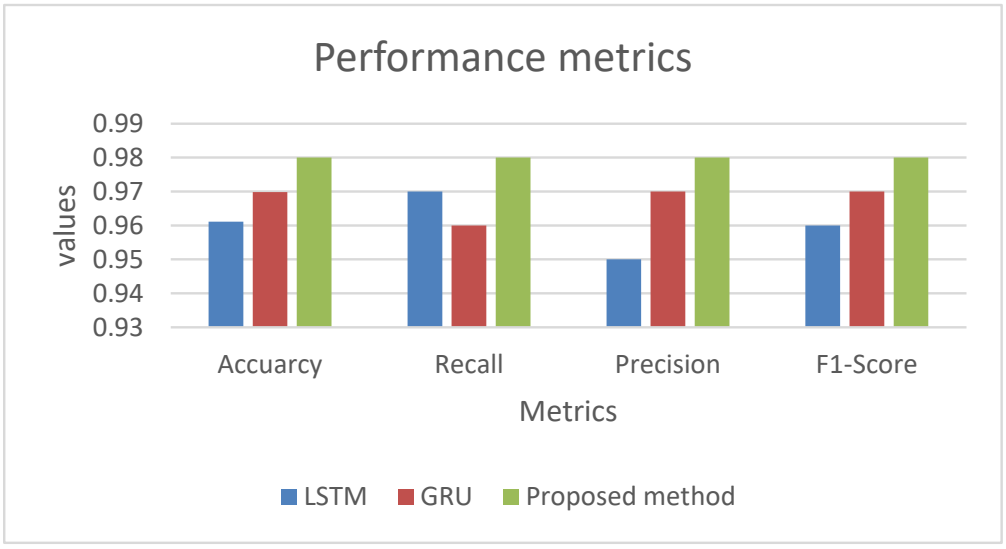



**Figure 16.** Comparative Analysis of Projected Model with Classical Method.

It is identified that the proposed model attained higher results by 0.0102 than GRU and by 0.0189 than the LSTM model in terms of accuracy. Likewise, it accomplished higher results in all the metrics, which shows the greater performance of the respective research. Table 9 and Figure 17 represent the overall comparative analysis of the presented system compared to classical models.

**Table 9.** Overall comparative analysis of projected method with conventional system.

| Methods | Accuarcy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| Supervised ML-RF, SVM and ANN. | 0.9737 | | | |
| Decision trees | | | | 0.86 |
| Naïve Bayes | | | | 0.824 |
| Support vector machine | | | | 0.755 |
| ANN | | | | 0.92018 |
| GA-DBN. | 0.8599 | | | |
| DBN | 0.82 | | | |
| CDBN | 0.8229 | | | |
| KG–DBN (Kullback Gaussian Deep Belief Network) | 0.8649 | | | |
| Logistic Regression | 0.8951 | 0.8 | 0.62 | 0.7004 |
| SVM | 0.9075 | 0.7011 | 0.6949 | 0.698 |
| Decision tree | 0.897 | 0.8541 | 0.6173 | 0.7166 |
| Auto-encoder | 0.822 | 1 | 0.4566 | 0.6269 |
| BAE 0 | 0.886 | 1 | 0.4063 | 0.5778 |
| BAE 1 | 0.90002 | 1 | 0.8198 | 0.901 |
| BAE 2 | 0.8425 | 0.8229 | 1 | 0.955 |
| BAE | 0.8761 | 0.8649 | 0.742 | 0.8113 |
| Proposed method | 0.98 | 0.98 | 0.98 | 0.98 |

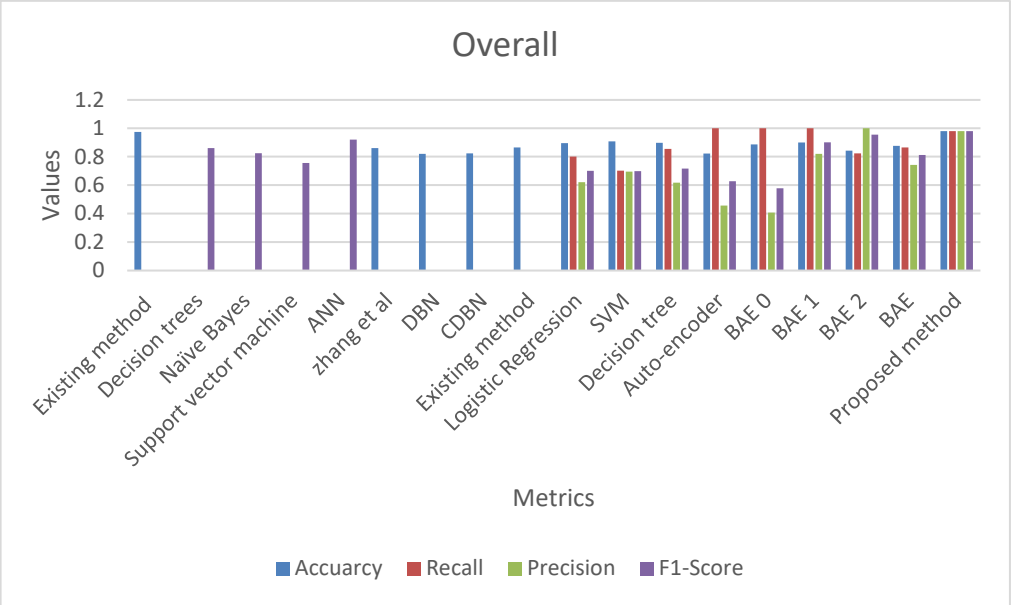



**Figure 17.** Overall comparative analysis of proposed model with Traditional Method.

Different measurements, such as accuracy, recall, precision, and F1-score, are employed to assess the effectiveness of the techniques in activities like classification or prediction. A comparison table displays the metrics of various methods in comparison to an established method that attains an accuracy of 0.9737. Decision trees, Naïve Bayes, SVM, and ANN were assessed, and ANN achieved the highest accuracy of 0.92018. Nonetheless, it is crucial to take into account additional measurements, such as precision and recall, for a thorough evaluation. The suggested approach stands out due to its accuracy of 0.98, as well as its superior recall, precision, and F1-score, surpassing both the current method and other sophisticated techniques. The comparison shown in Figure 17 and outlined in Table 9 clearly indicates that the suggested model surpasses conventional techniques by

a considerable amount. This excellent performance highlights the efficiency and possible practical value of the suggested method in real-life scenarios.

### *4.5. Discussion*

Several Conventional models thrived to accomplish network attack detection in diverse techniques. In the conventional model, IDS is processed with a UNSW–NB15 data set for constructing an integration classification system [53]. Similarly, the DL based model is used for cyber-attack and anomaly detection [59]. Accordingly, accuracy is an important metric that defines the whole performance of the detection model. Conversely, several models were lacking in the accuracy metric [27,43,50,52,53]. Moreover, the respective research attained better efficiency with two diverse datasets, which signifies greater performance of the presented system. The proposed system tackles the issue with an attained accuracy value of 98%. Experimental findings demonstrate that the suggested MLP technique presents a notable enhancement in the rate of accurate identification of malicious attacks. The enhancement stems from incorporating Random Information Grove Blend and weight weave layers alongside MLP layers. As per the analysis of the experiments, the proposed MLP algorithm yields more successful outcomes compared to alternative techniques, such as LR, SVM, DT, NB, ANN, and others. The suggested MLP offers an augmented accuracy in examining multi-class and binary attacks. It outperformed other algorithms regarding accuracy, precision, recall, F1 score, and test metrics in a broader evaluation. Compared to cutting-edge techniques, the proposed MLP method demonstrates its effectiveness as an IDS model. Consequently, it surpasses state-of-the-art intrusion detection techniques. Notably, the proposed MLP approach exhibits enhanced classification accuracy, distinguishing it from other methods and highlighting its value. Future endeavors aim to refine the classification accuracy further. Although the MLP model showed excellent performance, achieving 0.98 accuracy for multi-class classification for the UNSW–NB15 dataset and 100% accuracy for binary classification on the Scapy-generated dataset, there were no specific instances where the proposed model underperformed in the classification.

## 5. Conclusions

Cyber-attacks are a promptly emerging danger and an important hazard for worldwide security. In consequence, the proposed system aims to address the problem directly through conducting experiments in detection and safeguarding networks against the mischievous acts. In addition, it utilizes the UNSW–NB15 dataset and a manually generated Scapy Python (2.4.3) dataset for attack detection. Subsequently, the datasets undergo several data cleaning and preprocessing techniques in attaining the greatest level of accurateness against the biggest false alarm rate. The datasets are then split into train and test sets, with several experimentations conducted to estimate the efficiency of the exploration. Afterward, classification is carried out using the proposed MLP model, and the effectiveness is assessed using various performance metrics. Through experimentation with the classification approach, the presented system demonstrates that the proposed MLP model achieves an accuracy of 98%, surpassing other recently published models. The results unequivocally establish that the proposed model is an effective method for identifying cyber-attacks as well as securing networks. Generally, the proposed system delivers valued perceptions, within the potential of machine learning. by means of a tool for protection from cyber-attacks and safeguarding networks. Though exciting accuracy with F1-score outcomes were attained in the experimentation, there is still the possibility of development and additional optimization. Accordingly, the IDS of the presented system may vary in diverse network environments and its efficiency can be influenced on the basis of the network complexity. In future, examination of the proposed system in diverse types of networks and scenarios can enhance the efficiency. In addition, integrating advanced algorithms and techniques can be utilized to improve the accuracy of the respective method. Moreover, assessment of the generalization capabilities of the model by testing it on a variety of network attack datasets, and the integration of datasets containing a wider variety of uncommon or new

attack forms to evaluate how well the model can identify evolving cyber, is planned for the future direction of the proposed approach for enhancing efficacy.

**Funding:** This project is funded by the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, under project number (IF2/PSAU/2022/01/22846).

**Data Availability Statement:** The data will be made available by the authors on request.

**Acknowledgments:** The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work through the project number (IF2/PSAU/2022/01/22846).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. More, S.; Idrissi, M.; Mahmoud, H.; Asyhari, A.A.T. Enhanced Intrusion Detection Systems Performance With Unsw-Nb15 Data Analysis. *Algorithms* **2024**, *17*, 64. [CrossRef]
2. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. Igrf-Rfe: A Hybrid Feature Selection Method for Mlp-Based Network Intrusion Detection on Unsw-Nb15 Dataset. *J. Big Data* **2023**, *10*, 15. [CrossRef]
3. Drewek-Ossowicka, A.; Pietrołaj, M.; Rumiński, J. A Survey of Neural Networks Usage for Intrusion Detection Systems. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 497–514. [CrossRef]
4. Zhu, J.; Jang-Jaccard, J.; Singh, A.; Welch, I.; Harith, A.-S.; Camtepe, S. A Few-Shot Meta-Learning Based Siamese Neural Network Using Entropy Features for Ransomware Classification. *Comput. Secur.* **2022**, *117*, 102691. [CrossRef]
5. Alavizadeh, H.; Alavizadeh, H.; Jang-Jaccard, J. Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection. *Computers* **2022**, *11*, 41. [CrossRef]
6. Liu, T.; Sabrina, F.; Jang-Jaccard, J.; Xu, W.; Wei, Y. Artificial Intelligence-Enabled Ddos Detection for Blockchain-Based Smart Transport Systems. *Sensors* **2021**, *22*, 32. [CrossRef] [PubMed]
7. Wei, Y.; Jang-Jaccard, J.; Sabrina, F.; Singh, A.; Xu, W.; Camtepe, S. Ae-Mlp: A Hybrid Deep Learning Approach for Ddos Detection and Classification. *IEEE Access* **2021**, *9*, 146810–146821. [CrossRef]
8. Behiry, M.H.; Aly, M. Cyberattack Detection in Wireless Sensor Networks Using a Hybrid Feature Reduction Technique with Ai and Machine Learning Methods. *J. Big Data* **2024**, *11*, 16. [CrossRef]
9. Malik, M.; Ghous, H.; Mubeen, M.; Munir, A.M.; Ahmad, N. Intelligent Intrusion Detection System for Internet of Things Using Machine Learning Techniques. *Int. J. Inf. Syst. Comput. Technol.* **2024**, *3*, 23–39. [CrossRef]
10. Cengiz, K.; Lipsa, S.; Dash, R.K.; Ivković, N.; Konecki, M. A Novel Intrusion Detection System Based on Artificial Neural Network and Genetic Algorithm with a New Dimensionality Reduction Technique for Uav Communication. *IEEE Access* **2024**, *12*, 4925–4937. [CrossRef]
11. Kumar, P.; Kumar, A.A.; Sahayakingsly, C.; Udayakumar, A. Analysis of Intrusion Detection in Cyber Attacks Using Deep Learning Neural Networks. *Peer-Peer Netw. Appl.* **2021**, *14*, 2565–2584. [CrossRef]
12. Luo, C.; Tan, Z.; Min, G.; Gan, J.; Shi, W.; Tian, Z. A Novel Web Attack Detection System for Internet of Things Via Ensemble Classification. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5810–5818. [CrossRef]
13. Tekerek, A. A Novel Architecture for Web-Based Attack Detection Using Convolutional Neural Network. *Comput. Secur.* **2021**, *100*, 102096. [CrossRef]
14. Xuan, C.D.; Dao, M.H. A Novel Approach for Apt Attack Detection Based on Combined Deep Learning Model. *Neural Comput. Appl.* **2021**, *33*, 13251–13264. [CrossRef]
15. Sun, H.; Chen, M.; Weng, J.; Liu, Z.; Geng, G. Anomaly Detection for in-Vehicle Network Using Cnn-Lstm with Attention Mechanism. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10880–10893. [CrossRef]
16. Tang, D.; Tang, L.; Shi, W.; Zhan, S.; Yang, Q. Mf-Cnn: A New Approach for Ldos Attack Detection Based on Multi-Feature Fusion and Cnn. *Mob. Netw. Appl.* **2021**, *26*, 1705–1722. [CrossRef]
17. Zhang, H.; Li, Y.; Lv, Z.; Sangaiah, A.K.; Huang, T. A Real-Time and Ubiquitous Network Attack Detection Based on Deep Belief Network and Support Vector Machine. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 790–799. [CrossRef]
18. Khan, M.A. Hcrnnids: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. *Processes* **2021**, *9*, 834. [CrossRef]
19. Shitharth, S.; Prasad, K.M.; Sangeetha, K.; Kshirsagar, R.; Babu, T.S.; Alhelou, H.H. An Enriched Rpco-Bcnn Mechanisms for Attack Detection and Classification in Scada Systems. *IEEE Access* **2021**, *9*, 156297–156312. [CrossRef]
20. Oliveira, N.; Praça, I.; Maia, E.; Sousa, O. Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems. *Appl. Sci.* **2021**, *11*, 1674. [CrossRef]
21. Kravchik, M.; Shabtai, A. Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks And Pca. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 2179–2197. [CrossRef]
22. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated Ddos Attack Detection in Software Defined Networking. *J. Netw. Comput. Appl.* **2021**, *187*, 103108. [CrossRef]

23. Al-Haija, Q.A.; Zein-Sabatto, S. An Efficient Deep-Learning-Based Detection and Classification System for Cyber-Attacks in Iot Communication Networks. *Electronics* **2020**, *9*, 2152. [CrossRef]
24. Chen, D.; Yan, Q.; Wu, C.; Zhao, J. Sql Injection Attack Detection and Prevention Techniques Using Deep Learning. *J. Phys. Conf. Ser.* **2021**, *1757*, 012055. [CrossRef]
25. Kshirsagar, P.R.; Yadav, R.K.; Patil, N.N. Intrusion Detection System Attack Detection and Classification Model with Feed-Forward Lstm Gate in Conventional Dataset. *Mach. Learn. Appl. Eng. Educ. Manag.* **2022**, *2*, 20–29.
26. Alshingiti, Z.; Alaqel, R.; Al-Muhtadi, J.; Haq, Q.E.U.; Saleem, K.; Faheem, M.H. A Deep Learning-Based Phishing Detection System Using Cnn, Lstm, Lstm-Cnn. *Electronics* **2023**, *12*, 232. [CrossRef]
27. Salmi, S.; Oughdir, L. Cnn-Lstm Based Approach for Dos Attacks Detection in Wireless Sensor Networks. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 0130497. [CrossRef]
28. Pawar, M.V. Detection and Prevention of Black-Hole and Wormhole Attacks in Wireless Sensor Network Using Optimized Lstm. *Int. J. Pervasive Comput. Commun.* **2023**, *19*, 124–153. [CrossRef]
29. Krishnan, S.A.; Sabu, A.N.; Sajan, P.; Sreedeep, A. Sql Injection Detection Using Machine Learning. *Rev. Geintec-Gest. Inov. E Tecnol.* **2021**, *11*, 11.
30. Falor, A.; Hirani, M.; Vedant, H.; Mehta; Krishnan, D. A Deep Learning Approach for Detection of Sql Injection Attacks Using Convolutional Neural Networks. In *Proceedings of Data Analytics and Management Icdam 2021*; Springer: Singapore, 2022; Volume 2, pp. 293–304.
31. Tang, P.; Qiu, W.; Huang, Z.; Lian, H.; Liu, G. Detection of Sql Injection Based on Artificial Neural Network. *Knowl.-Based Syst.* **2020**, *190*, 105528. [CrossRef]
32. Akhtar, M.S.; Feng, T. Detection of Malware by Deep Learning as Cnn-Lstm Machine Learning Techniques in Real Time. *Symmetry* **2022**, *14*, 2308. [CrossRef]
33. Almomani, I.; Alkhayer, A.; El-Shafai, W. An Automated Vision-Based Deep Learning Model for Efficient Detection of Android Malware Attacks. *IEEE Access* **2022**, *10*, 2700–2720. [CrossRef]
34. Ariyadasa, S.; Fernando, S.; Fernando, S. Detecting Phishing Attacks Using a Combined Model of Lstm and Cnn. *Int. J. Adv. Appl. Sci* **2020**, *7*, 56–67.
35. Adebowale, M.A.; Lwin, K.T.; Hossain, M.A. Intelligent Phishing Detection Scheme Using Deep Learning Algorithms. *J. Enterp. Inf. Manag.* **2023**, *36*, 747–766. [CrossRef]
36. Dora, V.R.S.; Lakshmi, V.N. Optimal Feature Selection with Cnn-Feature Learning for Ddos Attack Detection Using Meta-Heuristic-Based Lstm. *Int. J. Intell. Robot. Appl.* **2022**, *6*, 323–349. [CrossRef]
37. Setitra, M.A.; Fan, M.; Agbley, B.L.Y.; Bensalem, Z.E.A. Optimized Mlp-Cnn Model to Enhance Detecting Ddos Attacks in Sdn Environment. *Network* **2023**, *3*, 538–562. [CrossRef]
38. Ma, Y.; Wu, L.; Li, Z. A Novel Face Presentation Attack Detection Scheme Based on Multi-Regional Convolutional Neural Networks. *Pattern Recognit. Lett.* **2020**, *131*, 261–267. [CrossRef]
39. Desta, A.K.; Ohira, S.; Arai, I.; Fujikawa, K. Rec-Cnn: In-Vehicle Networks Intrusion Detection Using Convolutional Neural Networks Trained on Recurrence Plots. *Veh. Commun.* **2022**, *35*, 100470. [CrossRef]
40. Gudla, S.P.K.; Bhoi, S.K. Mlp Deep Learning-Based Ddos Attack Detection Framework for Fog Computing. In *Advances in Distributed Computing and Machine Learning: Proceedings of Icadcml 2022*; Springer: Singapore, 2022; pp. 25–34.
41. Krithivasan, K.; Pravinraj, S.; Vs, S.S. Detection of Cyberattacks in Industrial Control Systems Using Enhanced Principal Component Analysis and Hypergraph-Based Convolution Neural Network (Epca-Hg-Cnn). *IEEE Trans. Ind. Appl.* **2020**, *56*, 4394–4404.
42. Zhang, G.; Li, J.; Bamisile, O.; Xing, Y.; Cao, D.; Huang, Q. Identification and Classification for Multiple Cyber Attacks in Power Grids Based on the Deep Capsule Cnn. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106771. [CrossRef]
43. Moghanian, S.; Saravi, F.B.; Javidi, G.; Sheybani, E.O. Goamlp: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm. *IEEE Access* **2020**, *8*, 215202–215213. [CrossRef]
44. Anand, A.; Rani, S.; Anand, D.; Aljahdali, H.M.; Kerr, D. An Efficient Cnn-Based Deep Learning Model to Detect Malware Attacks (Cnn-Dma) in 5g-Iot Healthcare Applications. *Sensors* **2021**, *21*, 6346. [CrossRef]
45. Elsayed, M.S.; Le-Khac, N.-A.; Albahar, M.A.; Jurcut, A. A Novel Hybrid Model for Intrusion Detection Systems in Sdns Based on Cnn and a New Regularization Technique. *J. Netw. Comput. Appl.* **2021**, *191*, 103160. [CrossRef]
46. Kaushik, P. Unleashing the Power of Multi-Agent Deep Learning: Cyber-Attack Detection in Iot. *Int. J. Glob. Acad. Sci. Res.* **2023**, *2*, 15–29. [CrossRef]
47. Issa, A.A.; Albayrak, Z. Ddos Attack Intrusion Detection System Based on Hybridization of Cnn and Lstm. *Acta Polytech. Hung.* **2023**, *20*, 105–123. [CrossRef]
48. Liu, G.; Zhang, J. CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 4705982. [CrossRef]
49. Yue, C.; Wang, L.; Wang, D.; Duo, R.; Nie, X. An Ensemble Intrusion Detection Method for Train Ethernet Consist Network Based on Cnn and Rnn. *IEEE Access* **2021**, *9*, 59527–59539. [CrossRef]
50. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. Cnn-Based Network Intrusion Detection Against Denial-of-Service Attacks. *Electronics* **2020**, *9*, 916. [CrossRef]

51. Najar, A.A.; Naik, S.M. Ddos Attack Detection Using Mlp and Random Forest Algorithms. *Int. J. Inf. Technol.* **2022**, *14*, 2317–2327. [CrossRef]
52. Muhuri, P.S.; Chatterjee; Yuan, X.; Roy, K.; Esterline, A. Using a Long Short-Term Memory Recurrent Neural Network (Lstm-Rnn) to Classify Network Attacks. *Information* **2020**, *11*, 243. [CrossRef]
53. Kumar, V.; Sinha, D.; Das, A.K.; Pandey, S.C.; Goswami, R.T. An Integrated Rule Based Intrusion Detection System: Analysis on Unsw-Nb15 Data Set and the Real Time Online Dataset. *Clust. Comput.* **2020**, *23*, 1397–1418. [CrossRef]
54. Almarshdi, R.; Nassef, L.; Fadel, E.; Alowidi, N. Hybrid Deep Learning Based Attack Detection for Imbalanced Data Classification. *Intell. Autom. Soft Comput.* **2023**, *35*, 297. [CrossRef]
55. Ahmad, M.; Riaz, Q.; Zeeshan, M.; Tahir, H.; Haider, S.A.; Khan, M.S. Intrusion Detection in Internet of Things Using Supervised Machine Learning Based on Application and Transport Layer Features Using Unsw-Nb15 Data-Set. *Eurasip J. Wirel. Commun. Netw.* **2021**, *2021*, 10. [CrossRef]
56. Han, H.; Kim, H.; Kim, Y. An Efficient Hyperparameter Control Method for a Network Intrusion Detection System Based on Proximal Policy Optimization. *Symmetry* **2022**, *14*, 161. [CrossRef]
57. Tian, Q.; Han, D.; Li, K.-C.; Liu, X.; Duan, L.; Castiglione, A. An Intrusion Detection Approach Based on Improved Deep Belief Network. *Appl. Intell.* **2020**, *50*, 3162–3178. [CrossRef]
58. Wang, D.; Nie, M.; Chen, D. Bae: Anomaly Detection Algorithm Based on Clustering and Autoencoder. *Mathematics* **2023**, *11*, 3398. [CrossRef]
59. Dutta, V.; Choraś, M.; Pawlicki, M.; Kozik, R. A deep learning ensemble for network anomaly and cyber-attack detection. *Sensors* **2020**, *20*, 4583. [CrossRef]